# Universitatea POLITEHINICA București Facultatea de Automatică și Calculatoare

Managementul Proiectelor Software

Hierarchy Analysis Software Development Document v 1.0

Echipa Farik

# Cuprins

Cuprins	2
1. Scopul documentului	
2. Vedere de ansablu a solutiei proiectate	3
TODO [ALL] Vom completa dupa finalizare	ea documentului

# 1. Scopul documentului

Acest document are rolul de a descrie soluția tehnică proiectată pentru componenta *Hierachy Analysis* din cadrul sistemului OCR modular pe care studenții îl dezvoltă în cadrul cursului de Managementul Proiectelor Software din cadrul Facultății de Automatică și Calculatoare.

Documentul serveşte drept ghid pentru implementarea soluţiei software pentru modulul *Hierarchy Analysis*.

# 2. Vedere de ansamblu a solutiei proiectate

## 2.1. Fond

Functiile si scopul acestui proiect sunt redactate in Documentul Specificatiilor de Proiect la adresa <a href="http://elf.cs.pub.ro/mps/wiki/proiect/descriere">http://elf.cs.pub.ro/mps/wiki/proiect/descriere</a>.

# 2.2. Context si design de baza

TODO [ALL] Vom specifica doar daca vom considera necesar care este flow-ul principal, eventual si o diagrama.

# 2.3. Decizii de implementare

TODO [DEV] Vom specifica care este limbajul folosit (java), pe ce platforma vom dezvolta, IDE-uri etc si in plus de ce s-a luat aceasta decizie.

Alte considerente: Java e free, fast, secure, reliable, cross platform etc.

## 2.4. Formatul fisierelor utilizate

Aplicatie foloseste fisiere XML ca modalitate de introducere a datelor in sistem. Datele de iesire sunt tot fisiere XML.

Toate fisierele XML cu care lucreaza componenta *Hierarchy Analysis* sunt validate de catre XSD-urile pus la dispozitie la aceasta adresa <a href="http://elf.cs.pub.ro/mps/wiki/proiect/specificatii-xsd">http://elf.cs.pub.ro/mps/wiki/proiect/specificatii-xsd</a>.

## 4. Modelul datelor:

**HierarchyAnalysis**: scopul acestei clase este obtinerea unui fisier XML care contine rezultatul analizei ierarhice. Ii permite utilizatorului sa si aleaga modul in care va obtine

fisierul de output XML. Prima varianta : incarca unul sau mai multe fisiere XML obtinute in urma analizei de layout si ruleaza un analizor ierarhic pentru acestea. In urma rularii va obtine fisierul cu rezultatul analizei ierarhice. Cea de a doua posibilitate este incarcarea directa a fisierului de catre utilizator si este cea recomandata de catre ALINA!!!!!!!

#### Structura clasei:

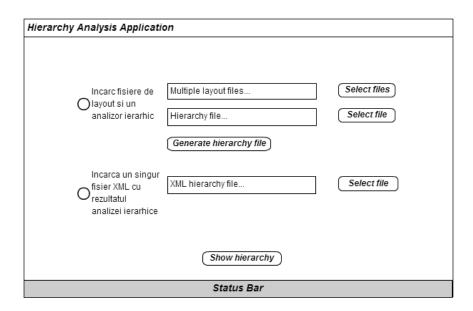
## Membrii clasei sunt urmatorii:

- Un grup de elemente RadioButton care permite utilizatorului selectia unui anumit mod de introducere a datelor. Exista 2 moduri posibile pentru a obtine fisierul de intrare :
  - 1. Selectia mai multor fisiere XML de layout si a unui analizor ierarhic. Pentru a selecta fisierele XML se va folosi JfileChooser-ul: BrowseXMLs, iar pentru selectia analizorului ierarhic se va folosi JfileChooser-ul: BrowseExec.
  - 2. Selectia unui singur fisier XML care contine rezultatul analizei ierarhice cu ajutorul unui JfileChooser numit : BrowseUniqueXML.

Validarea fisierelor XML de intrare se face dupa incarcarea acestora. Operatia este realizata in background, utilizatorul fiind instiintat daca operatia a avut succes cu ajutorul unui StatusBar, in care mesajele vor fi : Validare in curs, Fisiere XML valide sau Fisiere XML invalide.

- Un Jbutton : ShowHierarchy. La apasarea acestui buton se va inchide fereasta curenta si se va deschide o noua fereastra care va afisa ierarhia si va permite utilizatorului sa corecteze eventualele erori.

## Diagrama UI a clasei:



**Editor** : scopul acestei clase este afisarea diferitelor tipuri de blocuri, unirea lor , spargerea lor sau recatalogarea diferitelor blocuri.

## Structura clasei:

## Membrii clasei sunt urmatorii:

Clasa va contine doua Panel-uri. Cel din stanga va contine o suprafata reprezentand pagina, pe care se vor afisa blocurile de text si tipul acestora. In dreapta fiecarui bloc se vor afla 2 ComboBox-uri, unul pentru a permite editarea respectivului bloc : split, merge, schimbare tip si celalalt pentru schimbarea fontului blocului.

Panel-ul din dreapta va contine o suprafata reprezentand pagina, pe care se vor afisa imaginile ce au fost prelucrate de catre analizatorul de layout. Daca sunt inspectate mai multe imagini , acestea vor putea fi afisate una deasupra celelalte in panel ul acesta. Imaginile sunt afisate in cadrul unui JScrollPanel. De asemenea, exista 2 butoane care permit zoom-ul pe o anumita pagina. Pentru cazul in care zoom-ul este prea mare, ea va putea fi inspectata cu ajutorul unui JScrollBar orizontal.

In centrul ecranului se va afla un JscrollBar care va derula in mod sincron cele doua emisfere. In cazul unei desincronizari pe verticala, exista in partea dreapta un alt JscrollBar prin care se va face potrivirea.

Blocurile din emisfera stanga sunt instante ale unei clase parinte Element. Blocurile pot fi de tipul: Paragraph, Title, Subtitle, Article, Body, PageNumber, Image. Caracteristicile comune tuturor blocurilor sunt: existenta unei instante a clasei Cadru ( vizibila in interfata grafica printr un poligon incadrator ) si continutul propriu-zis de informatie.

TODO [QA]

Planul de testare

TODO [ALL]

Cerinte de sistem

procesor, RAM, runtime envs, Internet connection speed etc